HW01-因子有效性评价函数

22335047 郭振宇

2024 年 5 月 15 日

1 问题描述

任选一种因子有效性评价方式，通过代码实现成函数，函数输入为 dataframe，包含三列（日期、股票代码、因子值），函数返回为因子有效性分析的图表或数据。

2 问题分析

2.1 有效性评价方式

有效因子识别主要有三种方法:

1）分层法：每个截面日对个股按因子值排序，分层构建组合，每期调仓一次，观察各层组合表现。

2）回归法：将 T 期因子暴露与 T+1 期股票收益进行线性回归，回归系数即该因子的因子收益率。

3）IC 值法：计算第 T 期因子暴露与 T+1 期股票收益的相关系数。

函数实现过程中主要使用 IC 值法，同时结合分层法和回归法进行分析。

2.2 函数实现

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as st
import tushare as ts
import time
from scipy.stats import linregress, pearsonr, spearmanr
pro = ts.pro_api('129dd5438782d33a4e811764b69ed0dbc9fc0c4e53fa4ee4d3718f7c')

# 指定中文字体
plt.rcParams['font.sans-serif'] = ['Arial␣Unicode␣MS']
plt.rcParams['axes.unicode_minus'] = False  # 用来正常显示负号

# 定义函数
def factor_effectiveness_analysis(date, stock_code, factor):
    regress_coeff = []
    t_value = []
    ics = []
    for i in range(len(date)-1):
```

```
    ret = []
    # 获取每只股票周期截面的收益率
    for code in stock_code:
        df = pro.daily(ts_code=code, end_date=date[i+1], limit='10',
        fields='close')
        ret.append(df.close[len(df)-1]/df.close[0]-1)

    # 计算因子与收益率的相关性、线性回归系数、t统计量
    correlation,slope,t_statistic =calculate_statistics(factor[i], ret)
    ics.append(correlation)
    t_value.append(t_statistic)
    regress_coeff.append(slope)

# 计算平均IC值、IR值、负相关IC值占比、回归t检验均值
average_ics = np.mean(ics)
ir = np.mean(ics)/np.std(ics)
negative_ratio = len(list(filter(lambda x: x < 0, ics)))/len(ics)
regress_t_mean = np.mean(t_value)

IC = pd.DataFrame({'IC均值': [average_ics], 'ICIR值': [ir],
'负相关IC值占比': [negative_ratio], '回归t检验均值': [regress_t_mean]})

# 绘制IC值
plt.figure(figsize=(10, 6))
plt.plot(date[1:], ics, marker='o', linestyle='-')
plt.title('IC值随日期变化折线图')
plt.xlabel('日期')
plt.ylabel('IC值')
plt.ylim(-1, 1)
plt.xticks(rotation=90)
plt.savefig('IC-Date.png', dpi=300)
plt.show()
# 暂停以确保图表被渲染
plt.pause(0.1)
plt.close()

# 绘制因子收益率图表
plt.figure(figsize=(10, 6))
plt.plot(date[1:], regress_coeff, label='factor_return', color='green')
plt.xlabel('date')
plt.ylabel('(cum_sum) return')
plt.title('(累计) 因子收益率随日期变化折线图')

# 绘制因子收益率累积图表
```

```
    cumulative_sum = np.cumsum(regress_coeff)
    plt.plot(date[1:], cumulative_sum, label='cumulative_sum', color='blue')
    plt.legend(['因子收益率', '累计因子收益率'])
    plt.savefig('（累计）因子收益率随日期变化折线图.png', dpi=300)
    plt.show()

    return plt, IC


def calculate_statistics(data1, data2):
    # 计算相关系数
    correlation, _ = pearsonr(data1, data2)

    # 线性回归，计算回归的t统计量和相应的p值
    slope, intercept, r_value, p_value, std_err= linregress(data1, data2)
    t_statistic = slope / std_err
    p_value_t = 2 * (1 - st.t.cdf(abs(t_statistic), len(data1) - 2))

    return  correlation, slope, t_statistic
```

2.3 因子测试

选取“价量背离”因子进行函数测试。

$$\begin{aligned} Alpha &= -1 * CORR(VAWP, VOLUME, d) \\ &= -corr(vwap^{i}_{t-d:t}, volume^{i}_{t-d:t}) \end{aligned} \tag{1}$$

```
'''
以下为实际运行结果
'''
#  *Tu share* 获取数据参数设置
data = pro.daily(ts_code='000001.SZ', start_date='20161231', end_date=
'20181231', fields='trade_date')

date = []
for i in range(0, len(data.trade_date), 10):
    date.append(data.trade_date[i])
date = date[::-1]

# stock_code = list(df1['SecuCode'])
stock_code = ['000001.SZ','600000.SH','001872.SZ','300458.SZ','600200.SH']

factor = [[] for _ in range(len(stock_code))]
for index, code in enumerate(stock_code):
    factor[index] = []
```

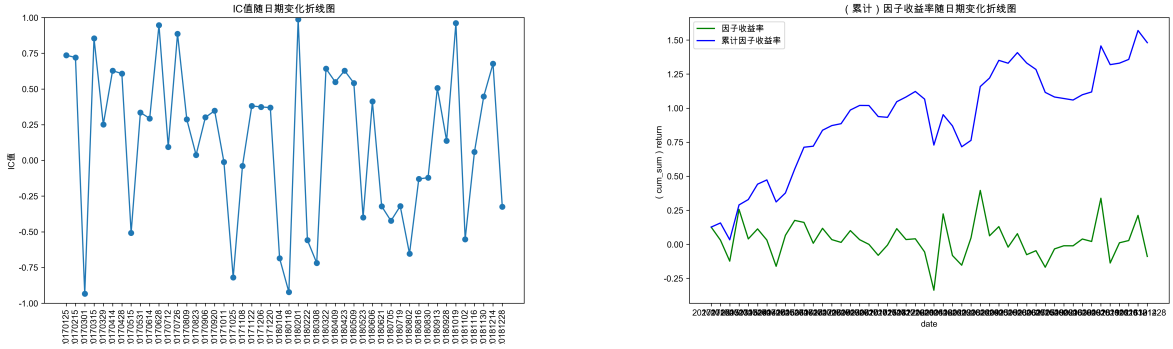
```
    for i in range(len(date)-1):
    df = pro.daily(ts_code=code, end_date=date[i], limit='10',
    fields='open,close,high,low,pre_close,change,pct_chg,vol,amount')
        factor[index].append(-np.corrcoef(df.vol, df.amount/df.vol)[0,1])
factor = np.array(factor).T

# 调用函数
plt, IC = factor_effectiveness_analysis(date, stock_code, factor)
plt.show()
print(IC)
```

2.4 结果输出

表 1: 因子有效性分析数据

因子	IC 均值	IR 值	负相关 IC 值占比	回归 t 检验均值
Alpha	0.136504	0.25189	0.375	0.545724



(a) IC 值随日期变化折线图

(b) （累计）因子收益率随日期变化折线图

图 1: 因子有效性分析图表

3 小注

1. 该因子的测试结果整体符合其定义，有效性分析的图表和数据输出可以根据实际情况进行调整。
2. 这里的局限性在于 Tu share 调取的数据有限，无限制可以研究更长日期、更高频率对应的结果。
3. 函数源代码见附件 factor_effectiveness_analysis.py，不同因子的分析只需修改因子的计算公式。